The DoD High Level Architecture and the Next Generation of DIS
(96-14-115)

Presented at the 14th Workshop on Standards for DIS

11 March 1996

Duncan C. Miller, Sc.D.

M.I.T. Lincoln Laboratory
244 Wood St.
Lexington, MA 02173-9108
617-981-7452
dmiller@ll.mit.edu

DoD Modeling and Simulation Management Directive 5000.59

- Signed by William J. Perry (then DepSecDef) 4 Jan 94
- Establishes DoD policy for the “strengthening, improvement, coordination, and management of modeling and simulation (M&S) within DoD”
- Establishes key management organizations
  - DoD Executive Council for M&S (EXCIMS)
  - Defense Modeling and Simulation Office
- Establishes requirements and responsibilities for
  - DoD M&S Master Plan
  - DoD M&S Investment Plan
  - DoD M&S Information Analysis Center
- DoD M&S Master Plan requires that all DoD simulations comply with a common High Level Architecture

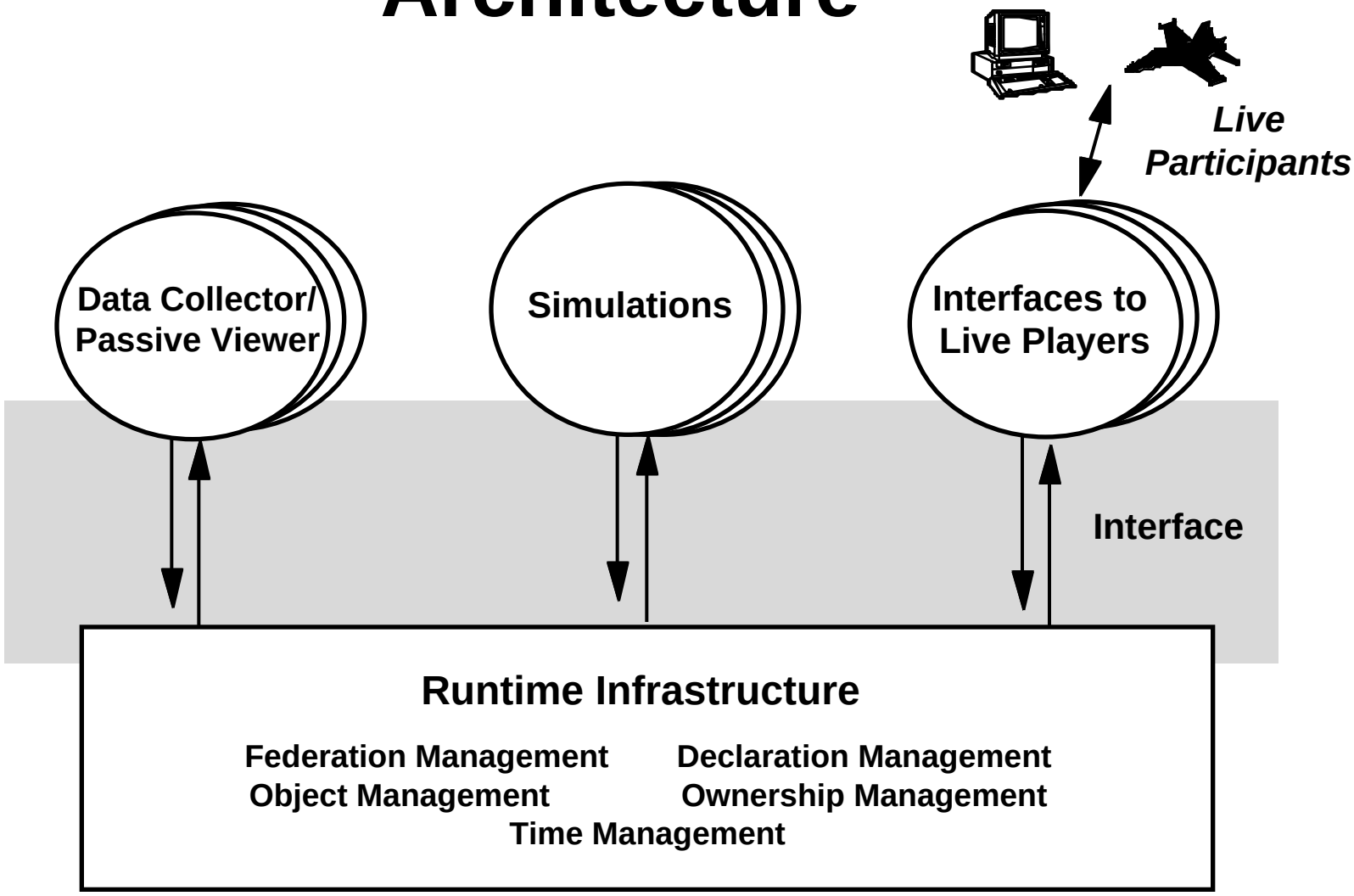
Duncan Miller
4/12/96

Rationale for HLA approach

- Basic premises:
  - No single simulation can satisfy the needs of all users
  - We can't anticipate all future uses of simulations and useful ways of combining simulations
  - Technology improvements will continue to occur
- Consequence:
  - Need a composable approach to constructing simulations
- Design principles:
  - Simulations should be built from modular components with well-defined functionality and interfaces
  - Simulation functionality should be separated from the general supporting infrastructure

Duncan Miller
4/12/96

Functional View of the Architecture



Duncan Miller
4/12/96

Why can’t the current DIS serve as the common architecture?

- To some extent, it can ... many of the ideas in the HLA came directly from the DIS community
- DIS has historically focused on continuous, real-time, man-in-the-loop simulations at the platform level
- Other categories of simulations use different time management approaches (e.g., event queues) and different levels of resolution (e.g., kilometers vs. meters) and granularity (e.g., aggregated combat units)
- We need to extend the DIS paradigm to cover these additional areas, while making DIS object models, data representations, and communication protocols more flexible and efficient

Duncan Miller
4/12/96

DIS++ concepts

- Object Interaction Protocols
- The Protocol Catalog
- Multiple data representations
- Communication protocols
- Attribute extrapolation algorithms
- Run-Time Infrastructure (RTI)
- Federation Object Model (FOM)
- Publishing and subscription
- Filtering and dynamic multicasting
- Forward/backward compatibility

The DIS standards deal with three conceptual levels of interactions:

- **Entity/object interaction protocols:** what information needs to be exchanged among simulations under what circumstances *(These are what most of the DIS community thinks of as "the PDUs.")*
- **Data representation formats:** how the state variables and other attributes of each entity are represented in these interactions
- **Communication protocols:** how the required information gets transmitted among the simulations *(These are what communications people think of as "the PDUs.)*

Entity/object interaction protocols

- The DIS++ Protocol Catalog will incorporate and extend the information defined in the current DIS protocols and PDU standards
- Additional protocols will be added as new phenomena and interactions are defined for new applications
- Proposed Object Interaction Protocols can be reviewed by the Standards Committee and added to the Protocol Catalog without reballoting the underlying standards

Multiple data representations

- The current DIS standards specify a fixed coordinate system, with fixed units and resolutions
- This system does not map well to some applications, such as instrumented ranges, which have their own coordinate systems and data standards
- DIS++ permits the use of alternative data representations from the Data Dictionary/Protocol Catalog
- If new data representations are required for an application, they can be added to the Data Dictionary/ Protocol Catalog without reballoting the underlying standards

Duncan Miller
4/12/96

Communications protocol

Current DIS	DIS ++
Each simulation is responsible for communicating its own entity state updates to other simulations	Each simulation communicates object attribute changes to the RTI, which acts as an intelligent delivery service
All entity state updates are transmitted to all simulations, whether they are interested or not	Attribute updates are delivered only to simulations that have declared interest in those updates
Each PDU contains a fixed set of state information	Only attribute changes are transmitted, as required
All entity state information is transmitted whenever any state variable change exceeds an agreed-upon threshold	Only changed attributes are transmitted

Duncan Miller
4/12/96

Attribute extrapolation

Current DIS	DIS ++
Only entity position and orientation are extrapolated	Any attribute can be extrapolated, if this is agreed to in the FOM
Eight dead-reckoning algorithms are prescribed in Annex B of IEEE 1278.1	Additional algorithms can be defined in the FOMs and added to the Protocol Catalog as required
Simulations are responsible for transmitting updates whenever the agreed uncertainty tolerance would be exceeded if it did not	Same; attribute updates are provided to the RTI for subscription-based distribution
Each simulation must apply the appropriate DR algorithm to extrapolate the attributes of each remote entity	Same; attribute extrapolation algorithms and parameters are established for each execution

Duncan Miller
4/12/96

Subscription and publishing

- Each simulation tells the RTI what classes of objects and which attributes of those objects it will represent
- Each simulation also tells the RTI what classes of objects and which attributes of those objects it needs to know about
- The RTI uses this information to set up potential communications paths
- Subscriptions are dynamic, and change frequently as an object moves, turns sensors on and off, etc.
- The RTI recalculates which attribute updates need to go where and establishes efficient groupings to do so

Filtering and dynamic multicasting

- RTI provides
  - object class and attribute filters
  - simple tests on selected attribute values
  - dynamic multicast grouping to ensure efficient delivery of relevant information
- Simulations (or “federates”) provide
  - more complex filter criteria, including those involving comparisons of attributes across objects
  - detailed sensor models for detection/identification
- Concept is a sequence of filters
  - multicast group assignments at the RTI level that minimize delivery of irrelevant information
  - more computationally expensive filters at the site or federate level, applied only to information that has already passed the initial screening tests

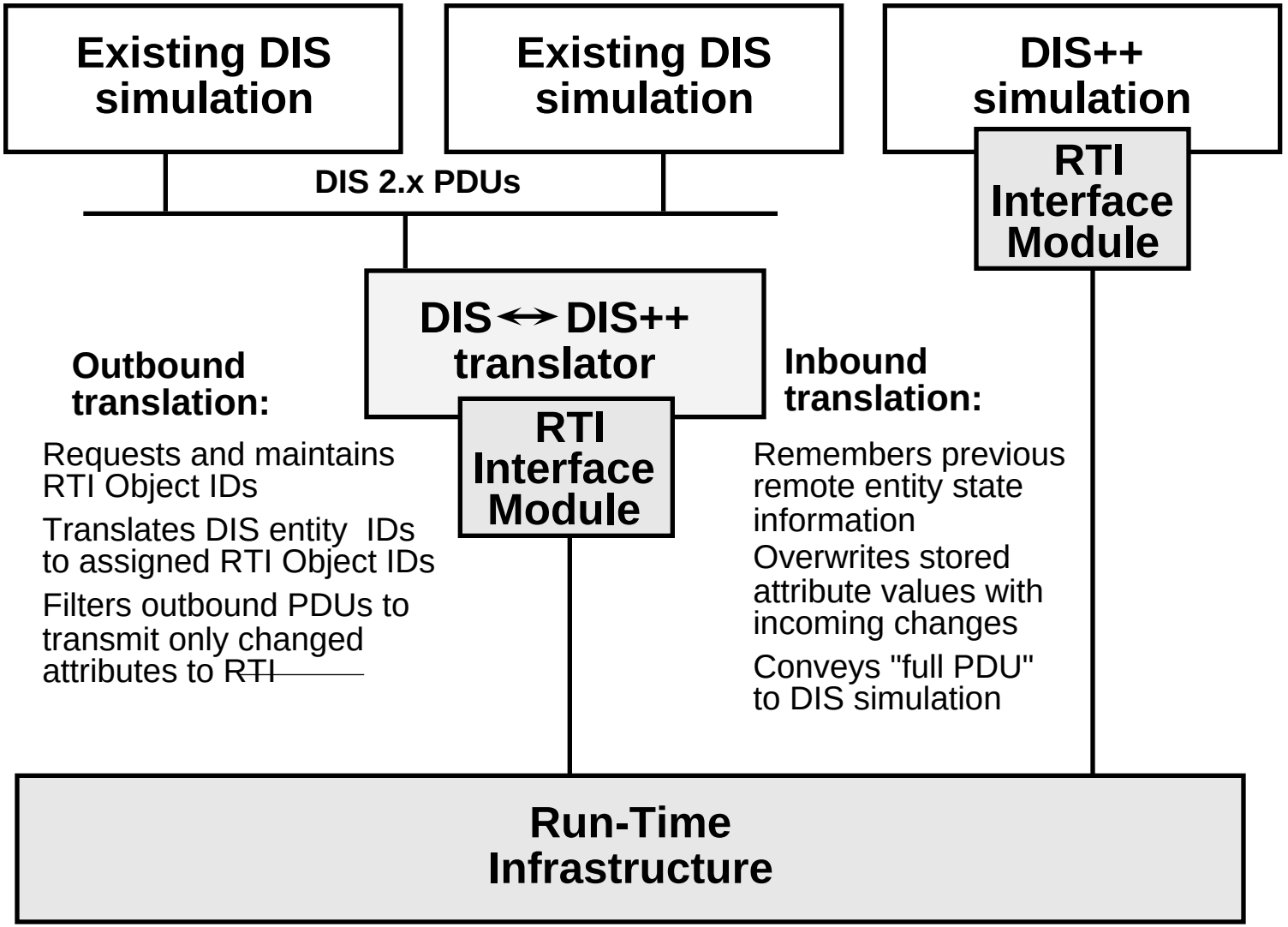
Exercise management

Function	Current DIS	DIS ++
Exercise creation	Establish an exercise ID	Create Federation Execution
Join an exercise	Listen in, start sending PDUs	Join Federation Execution
Assign object ID	Application creates a unique ID	Request Object ID from the RTI
Create an object	Create Entity	Instantiate Object
Discover new object	ESPDU arrives from unknown entity	Instantiate Discovered Object
Delete an object	Remove Entity	Delete Object
Leave an exercise	Stop sending PDUs	Resign Federation Execution

Backward and forward compatibility

- New DIS++ functionality:
  - explicit service calls for many functions that are only implicit in the current DIS
  - explicit time management services to insure proper delivery order of updates, when this is important
  - explicit ownership management services
  - transmission of updates only for changed attributes
- How can DIS++ functionality be incorporated in existing DIS simulations?
  - by directly implementing the RTI interface services in the application, thus making it DIS++ compliant
  - by incorporating a package of services developed by someone else (as many use VR-Link now)
  - by developing a separate “translator” application (*not a recommended approach, but a helpful way to visualize the issues involved in backward compatibility...*)

Conceptual interactions between DIS and DIS++ simulations



Duncan Miller
4/12/96